



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/878,859	06/11/2001	Simon Lok	AP33285	2657
21003	7590	12/30/2005	EXAMINER	
BAKER & BOTTS 30 ROCKEFELLER PLAZA NEW YORK, NY 10112			YIGDALL, MICHAEL J	
			ART UNIT	PAPER NUMBER
			2192	

DATE MAILED: 12/30/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/878,859

Applicant(s)

LOK ET AL.

Examiner

Michael J. Yigdal

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 01 September 2005 and 29 September 2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 01 September 2005 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This Office action is responsive to Applicant's submissions filed on September 1, 2005 and September 29, 2005. Claims 1-20 are pending.

Response to Amendment

2. It is noted that the amendment to the claims does not comply with the requirements of 37 CFR 1.121(c) because the status of claims 6-8 and 20 is not "original" as Applicant indicates (amendment, pages 3, 4 and 7), but is rather --currently amended--.

Response to Arguments

3. Applicant's arguments filed on September 1, 2005 have been fully considered but they are not persuasive.

Applicant contends that "the proposed combination of Muta and Mittal would require a substantial reconstruction and redesign of the elements shown in Muta as well as a change in the basic principle under which Muta was designed to operate" (remarks, page 10, last paragraph).

However, the examiner does not agree with Applicant's conclusion. Although Muta does disclose that the drawing function is performed at the server (see, for example, column 11, lines 7-21), Muta also discloses that the respective drawing function is performed at the remote client (see, for example, column 11, lines 50-62). In other words, the drawing function is performed at both the server and the remote client. As Applicant notes, one objective of Muta is to reduce the use of resources at the remote client (remarks, page 10, last paragraph). It is apparent, therefore, that the use of resources at the remote client is reduced even though the respective drawing function is performed at the remote client.

Mittal discloses an analogous method in which the drawing function is performed at the remote client, but here *only* at the remote client (see, for example, column 6, line 64 to column 7, line 1), as Applicant acknowledges (remarks, page 11, second paragraph). The remote client is able to control an applet at the server (see, for example, column 6, lines 32-38). One objective of Mittal is to reduce the use of time and memory resources at the remote client (see, for example, column 2, lines 34-43). In fact, Mittal discloses that the amount of data downloaded to the remote client is reduced even though the remote client is able to interact with the applet at the server (see, for example, column 7, lines 1-7).

The proposed combination of Muta and Mittal, to modify Muta such that the respective drawing function is performed only at the remote client instead of at both the server and the remote client, would not render Muta unsatisfactory for its intended purpose and would not change the principle of its operation. Muta already performs the respective drawing function at the remote client and reduces the use of resources at the remote client. Mittal performs the drawing function *only* at the remote client and likewise reduces the use of resources at the remote client. There is no suggestion that merely *not* performing the drawing function at the server in Muta would somehow change the principle of its operation. If Muta intends to perform more intensive functions at the server, as Applicant suggests (remarks, page 11, first paragraph), then it follows that Muta is amenable to the operation of Mittal, in which the functions of the applet are likewise performed at the server.

Applicant contends that neither Muta nor Cohen, either alone or in combination, disclose or suggest the step of “substituting said at least one portion of the code relevant to executing the

function with the portion of code configured to issue the remote command to execute the function” as recited in dependent claim 9 (remarks, page 12, last paragraph).

However, this argument was addressed in the Office action mailed on June 1, 2005. To reiterate, Muta discloses substituting or replacing the normal flow of the graphics engine 321 and the display driver 325 with a drawing command monitor 323 configured to issue the drawing command to the remote client to execute the function (see, for example, column 11, lines 7-21). The graphics engine 321 and the display driver 325 are “said at least one portion of the code relevant to executing the function.” The drawing command monitor 323 is “the portion of code configured to issue the remote command to execute the function.”

Thus, Muta teaches substituting or replacing “said at least one portion of the code relevant to executing the function” with “the portion of code configured to issue the remote command to execute the function.” It is considered a substitution because the window system 320 no longer consists of only the graphics engine 321 and the display driver 325; instead, the window system 320 is substituted with an arrangement that now includes the drawing command monitor 323 (see, for example, FIG. 8).

Muta discloses, “The drawing command monitor 323 is situated between the Graphics engine 321 and the display driver 325 of the window system 320, and can act as the display driver 325 with respect to the Graphics engine 321, while acting as the Graphics engine 321 with respect to the display driver 325” (column 11, lines 7-12). In other words, from the perspective of the graphics engine 321, the display driver 325 is substituted or replaced with the drawing command monitor 323, and from the perspective of the display driver 325, the graphics engine 321 is substituted or replaced with the drawing command monitor 323.

Double Patenting

4. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and, *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

5. Claims 1-20 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-20 of copending Application No. 10/296,193. Although the conflicting claims are not identical, they are not patentably distinct from each other because both recite a method for distributed processing through a server and a remote client and a corresponding distributed computer system having at least one server and one remote client.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 1-8 and 10-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,286,003 to Muta (art of record, "Muta") in view of U.S. Pat. No. 6,233,609 to Mittal (art of record, "Mittal").

With respect to claim 1 (currently amended), Muta discloses a method for distributed processing through a server and a remote client wherein an application is executed entirely in the server (see, for example, FIGS. 2 and 8, and column 1, lines 56-60), wherein the application is configured to interact with a user interface toolkit according to an application programming interface (see, for example, column 11, lines 13-21), and wherein the user interface toolkit has a component that performs a function (see, for example, column 2, lines 45-54), the method comprising:

(a) providing the user interface toolkit on the remote client comprising a component comprising code to perform the function on the remote client, wherein said component is related to user interaction, and to generate an event coupled to said component in response to user

Art Unit: 2192

interaction with said component (see, for example, master applet 215 in FIG. 8 and column 2, lines 36-44, which shows a user interface toolkit on the remote client having an event monitor, i.e. a component related to user interaction, and an event sender, i.e. for generating events coupled to the monitor in response to user interaction);

(b) providing a remote-capable user interface toolkit on the server comprising a remote-capable component which is configured to interact with the application according to the application programming interface and comprising code to generate a message to the component on the remote client to perform the respective function on the remote client (see, for example, slave daemon 247 in FIG. 8 and column 10, lines 56-67, which shows a remote-capable user interface toolkit on the server having an event analyzer component for interacting with an application by way of hooks and window messages, i.e. by way of an application programming interface);

(c) invoking the remote-capable user interface toolkit by the application to perform the function according to the application programming interface (see, for example, column 10, lines 56-67 and column 11, lines 13-21, which shows the application invoking the remote-capable user interface toolkit by calling the drawing functions of the application programming interface);

(d) generating the message by the remote-capable component of the remote-capable user interface toolkit on the server in response to the invocation by the application, the message comprising a higher-level command to the user interface toolkit on the remote client to perform the function (see, for example, column 11, lines 29-62, which shows generating messages, in the form of drawing commands, to send to the user interface toolkit on the remote client to perform the respective function, and see, for example, FIG. 8 and column 11, lines 13-21, which shows

that the drawing commands to the remote client are based on an application programming interface and are thus higher-level commands than the pixel data provided by the display driver 325 to the screen 245);

(e) communicating the message between the remote-capable user interface toolkit on the server and the user interface toolkit on the remote client (see, for example, column 11, lines 29-49, which shows sending or communicating the drawing command message between the server and the remote client); and

(f) performing the function on the remote client by the component of the user interface toolkit in response to the message (see, for example, column 11, lines 50-62, which shows the user interface toolkit performing the function on the remote client in response to the message).

Although Muta discloses that the respective function is performed at the remote client (see, for example, column 11, lines 50-62), Muta also discloses that the function is performed at the server (see, for example, column 11, lines 7-21), and therefore does not expressly disclose the limitation wherein the respective function is only performed at the remote client.

However, Mittal discloses an analogous method for distributed processing through a server and a remote client (see, for example, the abstract), wherein the functions of the user interface are performed only at the remote client (see, for example, client 133 in FIG. 3 and column 6, line 64 to column 7, line 7), so that a plurality of clients can interact with an application independently (see, for example, column 6, lines 32-38).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the method of Muta such that the respective function is only performed at

the remote client, as taught by Mittal, so that a plurality of clients can interact with the application independently.

With respect to claim 2 (original), Muta further discloses the limitation wherein the component in the user interface toolkit is configured to render a graphical item and the remote-capable component is configured to generate a message to render the graphical item (see, for example, drawing command analyzer 337 in FIG. 8, which shows the component in the user interface toolkit for rendering a graphical item, and drawing command sender 329 in FIG. 8, which shows the remote-capable component for generating the message), and wherein communicating the message between the remote-capable user interface toolkit on the server and the user interface toolkit on the remote client comprises transmitting the message to the user interface toolkit on the remote client to render the graphical item (see, for example, column 11, lines 29-49, which shows sending or transmitting the message to render the graphical item).

With respect to claim 3 (original), Muta further discloses the limitation wherein performing the function on the remote client by the component of the user interface toolkit comprises rendering the graphical item on the remote client in response to the message (see, for example, column 11, lines 50-62, which shows rendering the graphical item on the remote client in response to the drawing command message).

With respect to claim 4 (original), Muta further discloses the limitation wherein the component in the user interface toolkit is configured to install an event handler and the remote-capable component is configured to generate a message to install the event handler (see, for example, column 9, lines 26-35, which shows activating or installing an event monitor in

response to a notice or message from the server, and lines 36-52, which further shows using an event handler), and wherein communicating the message between the remote-capable user interface toolkit on the server and the user interface toolkit on the remote client comprises transmitting the message to the user interface toolkit on the remote client to install an event handler (see, for example, column 9, lines 6-17, which shows transmitting the message from the remote-capable user interface toolkit on the server to the user interface toolkit on the remote client).

With respect to claim 5 (original), Muta further discloses the limitation wherein performing the function on the remote client by the component of the user interface toolkit comprises installing the event handler on the remote client in response to the message (see, for example, column 9, lines 26-35, which shows activating or installing an event monitor on the remote client in response to the message).

With respect to claim 6 (currently amended), Muta further discloses:

(a) generating an event by the remote-capable component of the remote-capable user interface toolkit in response to the invoking (see, for example, column 10, lines 56-67 and column 11, lines 13-21, which shows invoking the remote-capable user interface toolkit, and lines 29-62, which further shows generating drawing commands or events to send to the user interface toolkit on the remote client); and

(b) wherein communicating the message between the remote-capable user interface toolkit on the server and the user interface toolkit on the remote client comprises asynchronously transmitting the event to the user interface toolkit (see, for example, FIG. 8, which shows

asynchronous communications between the remote-capable user interface toolkit on the server and the user interface toolkit on the client with drawing command buffer 327).

With respect to claim 7 (currently amended), although Muta discloses remotely controlling graphical applications on the server (see, for example, column 2, line 55 to column 3, line 4), which encompasses database searching applications, Muta does not expressly disclose the limitation wherein the application is a database searching application configured to search a database for information in response to a user-defined request,

(a) wherein generating an event by the remote-capable component of the remote-capable user interface toolkit comprises identifying information from the database in response to the user-defined request; and

(b) wherein asynchronously transmitting the event to the user interface toolkit comprises asynchronously transmitting a message to the remote client to render the information from the database identified in the step of generating an event.

However, as presented above, Mittal discloses an analogous method for distributed processing through a server and a remote client (see, for example, the abstract). Mittal discloses a database searching application for searching and identifying information from a database in response to a user-defined request and relaying that information to the client (see, for example, GUI 145, applet 141 and database 143 in FIG. 3, and column 7, lines 48-54). Mittal further discloses transmitting a message to the client to render that information (see, for example, column 7, lines 14-24). The method enables the client to interact with the application and database with minimal downloading (see, for example, column 7, lines 25-31), by not having to

download the application and database (see, for example, column 5, lines 3-7), so as to save time and memory resources (see, for example, column 2, lines 34-43).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the method of Muta such that the application is a database searching application configured to search a database for information in response to a user-defined request, wherein the step of generating an event by the remote-capable component of the remote-capable user interface toolkit comprises identifying information from the database in response to the user-defined request, and wherein the step of asynchronously transmitting the event to the user interface toolkit comprises asynchronously transmitting a message to the remote client to render the information from the database identified in the step of generating an event, as taught by Mittal. One of ordinary skill in the art would have been motivated to enable the client to interact with the application and the database with minimal downloading, by not having to download the application and database, so as to save time and memory resources.

With respect to claim 8 (currently amended), Mittal further discloses the limitation wherein the application is a web browser (see, for example, column 7, lines 8-13, which shows a web browser) and wherein the database is the World Wide Web (see, for example, database 143 and WAN 135 in FIG. 3, and column 5, lines 28-29, which shows that WAN 135 may be the Internet, and column 6, lines 39-41, which further shows that the knowledge base and thus the database are web-based, i.e. the World Wide Web),

(a) wherein identifying information from the database comprises identifying information from the World Wide Web (see, for example, database 143 and WAN 135 in FIG. 3, and column 5, lines 28-29, which shows that WAN 135 may be the Internet, and column 6, lines 39-41,

which further shows that the knowledge base and thus the database are web-based, i.e. the World Wide Web); and

(a) wherein asynchronously transmitting a command to the remote client to render the information from the database comprises asynchronously transmitting a command to the remote client to render the information from the World Wide Web (see, for example, database 143 and WAN 135 in FIG. 3, and column 5, lines 28-29, which shows that WAN 135 may be the Internet, and column 6, lines 39-41, which further shows that the knowledge base and thus the database are web-based, i.e. the World Wide Web).

With respect to claim 10 (currently amended), the limitations recited in the claim are analogous to those of claim 1 (see the rejection of claim 1 above). Muta further discloses a distributed computer system having a server (see, for example, slave server 240 in FIGS. 2 and 4) and a remote client (see, for example, master controller 210 in FIGS. 2 and 4).

With respect to claims 11 (original), 12 (currently amended), 14 (original) and 15 (currently amended), the limitations recited in the claims are analogous to those of claim 2 (see the rejection of claim 2 above).

With respect to claims 13 (original) and 16 (original), the limitations recited in the claims are analogous to those of claim 3 (see the rejection of claim 3 above).

With respect to claims 17 (original) and 18 (currently amended), the limitations recited in the claims are analogous to those of claim 4 (see the rejection of claim 4 above).

With respect to claim 19 (original), the limitations recited in the claim are analogous to those of claim 5 (see the rejection of claim 5 above).

With respect to claim 20 (currently amended), the limitations recited in the claim are analogous to those of claim 6 (see the rejection of claim 6 above).

8. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Muta in view of Mittal, as applied to claim 1 above, and further in view of U.S. Pat. No. 6,011,918 to Cohen et al. (art of record, "Cohen").

With respect to claim 9 (currently amended), although Muta discloses substituting the normal flow of the graphics engine and display driver, i.e. the portion of code relevant to executing the function, with a component that issues drawing commands to the remote client to execute the respective function (see, for example, column 11, lines 7-12), Muta does not expressly disclose the limitation wherein the step of providing a remote-capable user interface toolkit on the server further comprises:

(a) providing a code-generating computer program configured to read the code of a component of the user interface toolkit and to generate the remote-capable component of the remote-capable user interface toolkit by substituting at least one portion of the code relevant to executing the function with at least one portion of code configured to issue a remote command to execute the function;

(b) reading in the code of the component of the user interface toolkit;

(c) generating the remote-capable component of the remote-capable user interface toolkit by substituting said at least one portion of the code relevant to executing the function with said at least one portion of code configured to issue the remote command to execute the function.

However, Cohen discloses generating code automatically to distribute an application between a server and a remote client, by reading the code and substituting the components and functions with a remote-capable configuration (see, for example, column 3, lines 11-43), so as to support a plurality of computing topologies without having to redesign the distributed system for every case (see, for example, column 2, lines 62-64).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the distributed server and remote client system of Muta with the code generating features taught by Cohen, such that substituting the normal flow of the graphics engine and display driver, i.e. the portion of code relevant to executing the function, with a component that issues drawing commands to the remote client to execute the respective function, as disclosed by Muta (see, for example, column 11, lines 7-12), is performed by a code-generating computer program such as taught by Cohen. One of ordinary skill in the art would have been motivated to provide support for a plurality of computing topologies without having to redesign the distributed system for every case.

Conclusion

9. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



TUAN DAM
SUPERVISORY PATENT EXAMINER

Application/Control Number: 09/878,859
Art Unit: 2192

Page 17

MY

Michael J. Yigdall
Examiner
Art Unit 2192

mjy



8/8

```
public void registerDisplay (RJFrame d,  
    RJFCFactory f) throws RemoteException {  
  
    JTextArea TheArea = f.getRJTextArea (20, 20) ;  
    TheArea.addKeyListener (new  
        TextAreaKeyListener( ) ) ;  
  
    JScrollPane Pane = f.getRJScrollPane ( ) ;  
    Pane.setViewportViewView (textArea) ;  
  
    RJTextField StatusBar = f.getRJTextField ( ) ;  
    StatusBar.setEditable (false) ;  
  
    RContainer rc = d.getContentPane ( ) ;  
    rc.setLayout (new BorderLayout ( ) ) ;  
    rc.add (StatusBar, BorderLayout.SOUTH) ;  
    rc.add (CreateMenu ( ), BorderLayout.NORTH) ;  
    rc.add (Pane, BorderLayout.CENTER) ;  
  
}
```

FIG.9(a)

```
public MyJFrame () extends JFrame {  
  
    JTextArea TheArea = new JTextArea (20, 20) ;  
    TheArea.addKeyListener (new  
        TextAreaKeyListener( ) ) ;  
  
    JScrollPane Pane = new JScrollPane ( ) ;  
    Pane.setViewportViewView (textArea) ;  
  
    JTextField StatusBar = new TextField ( ) ;  
    StatusBar.setEditable (false) ;  
  
    Container c = this.getContentPane ( ) ;  
    c.setLayout (new BorderLayout ( ) ) ;  
    c.add (StatusBar, BorderLayout.SOUTH) ;  
    c.add (CreateMenu ( ), BorderLayout.NORTH) ;  
    c.add (Pane, BorderLayout.CENTER) ;  
  
}
```

FIG.9(b)
(PRIOR ART)